

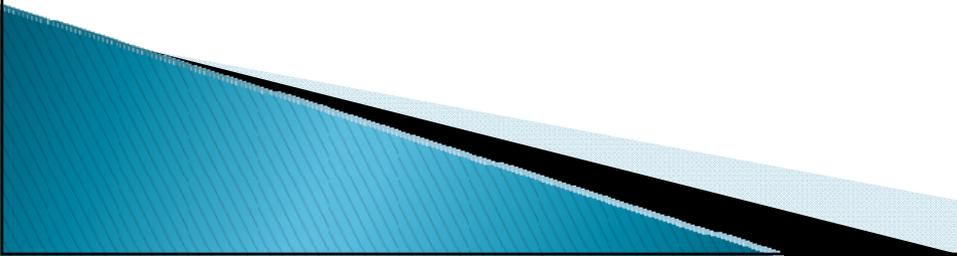
Трансляция описаний конечных автоматов, представленных в формате MS Visio, в исходный код на языке C

Леонид Столяров, 8 класс, лицей «Вторая школа», Москва

Научный руководитель: И. Р. Дединский, лицей «Вторая школа», Москва

Консультант: д.т.н., проф. А. А. Шалыто, ИТМО, Санкт-Петербург

ТРЕБОВАНИЯ К ПРОГРАММЕ

- ▶ Обеспечение эффективности отладки и тестирования автоматных программ
 - ▶ Повышение эффективности ручного и автоматического кодирования программ
 - ▶ Автоматическое документирование логики работы автоматной программы
- 

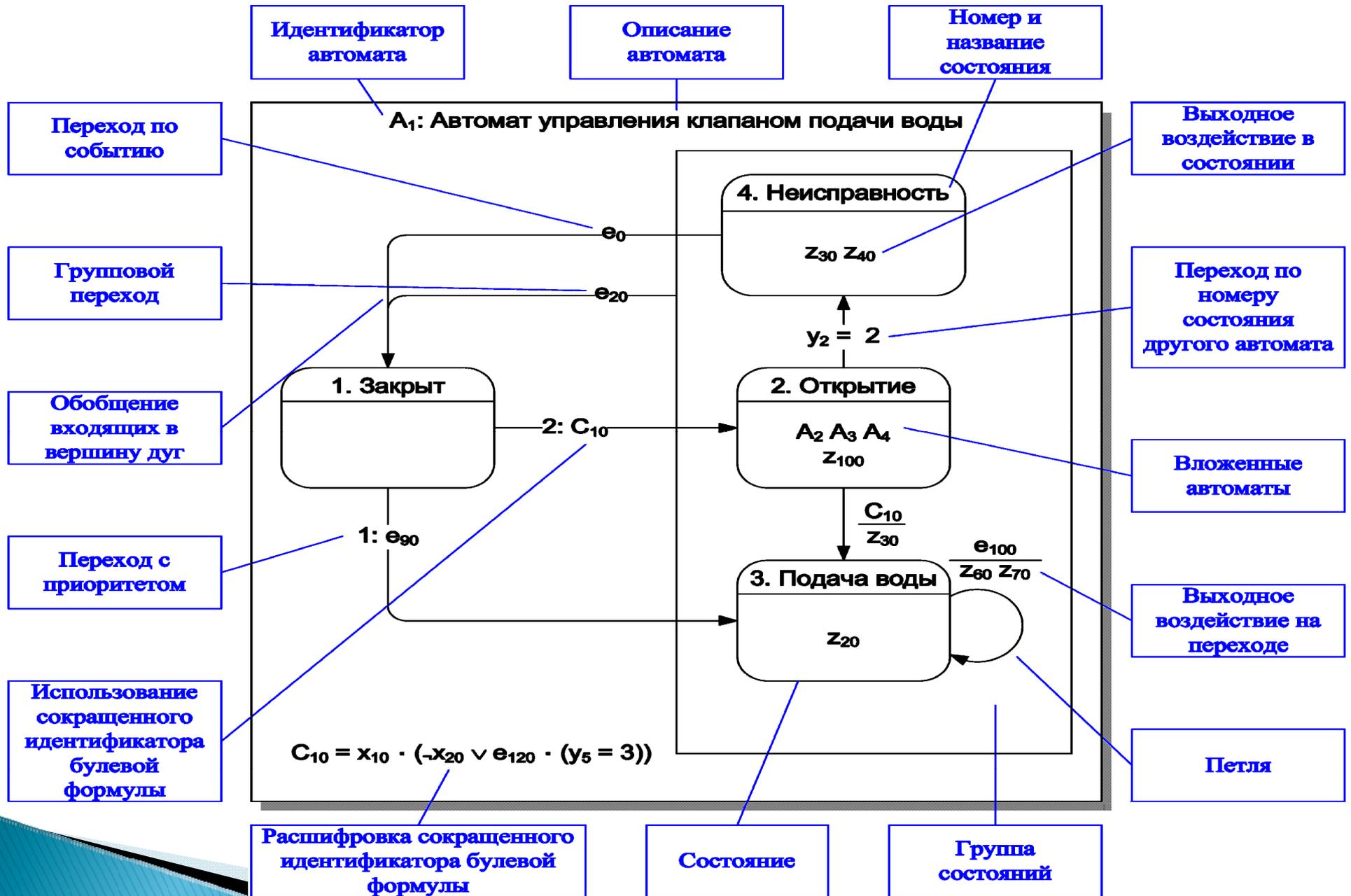
ЦЕЛЬ РАБОТЫ

Разработка программы – транслятора автоматных графов

ЗАДАЧИ

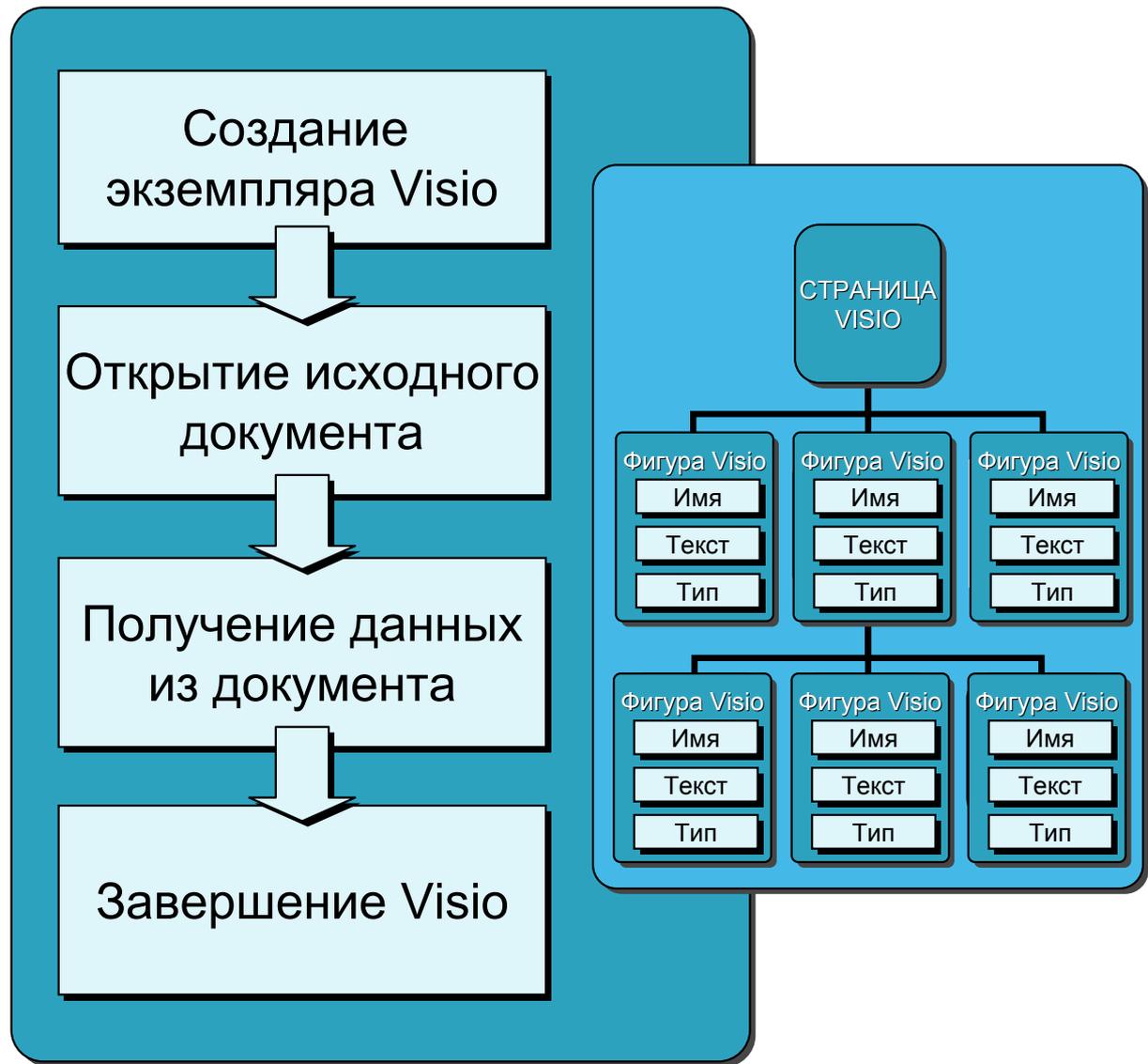
- ▶ Трансляция описаний автоматов, представленных в формате Visio, в код на С
- ▶ Сохранение исходных данных в файл XML
- ▶ Качественная диагностика ошибок в исходных данных
- ▶ Высокое качество и надёжность генерируемого кода
- ▶ Лёгкая интеграция в автоматические процессы сборки приложений

ФОРМАТ ОПИСАНИЯ АВТОМАТА



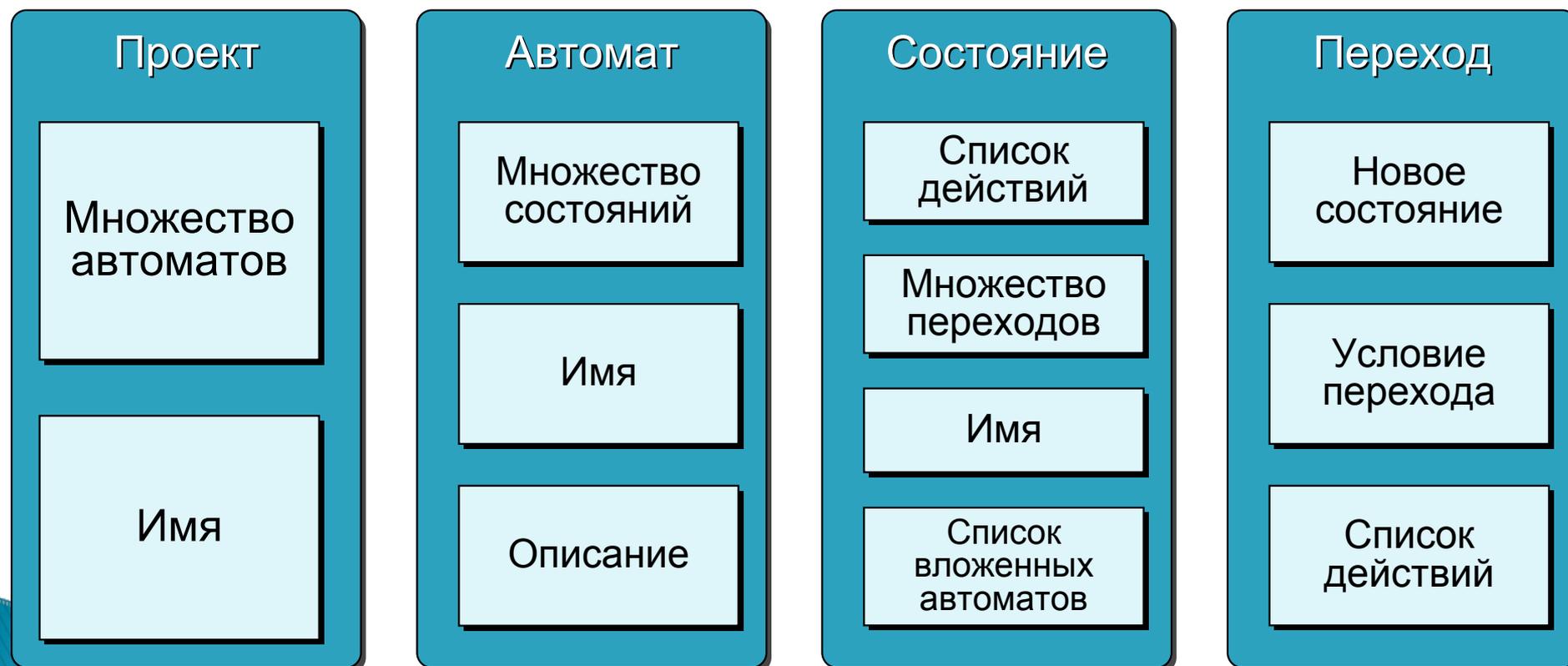
ВЗАИМОДЕЙСТВИЕ С MS VISIO

- ▶ Взаимодействие происходит через COM-объект Visio
- ▶ Внутренняя структура документа Visio – это иерархический набор фигур
- ▶ Импорт схемы из Visio заключается в обходе дерева фигур
- ▶ Фигуры элементов графа переходов загружаются из файла шаблонов Visio



ВНУТРЕННЕЕ ПРЕДСТАВЛЕНИЕ ДАННЫХ

- ▶ Внутренне исходные данные Visio представляются в виде модели, основанной на переходах и состояниях.



АЛГОРИТМ ТРАНСЛЯЦИИ

Чтение данных из исходного документа



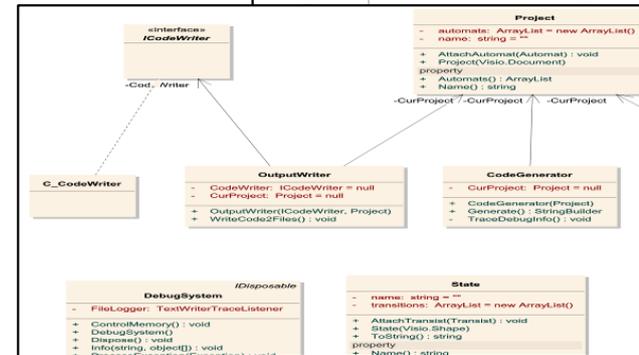
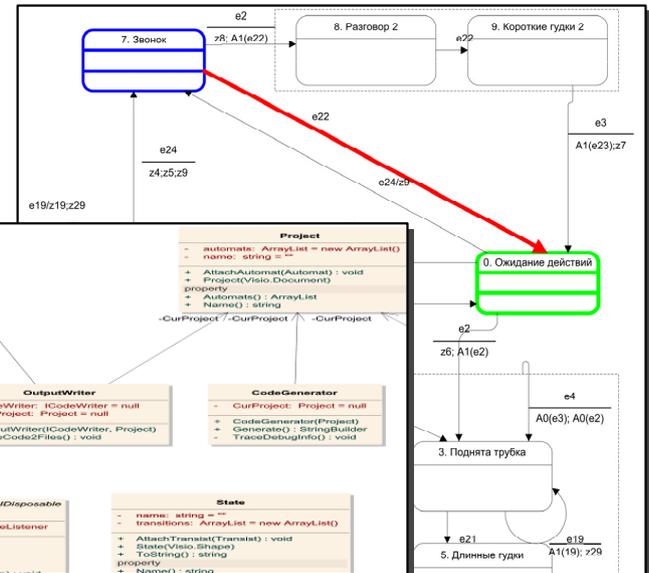
Преобразование исходных данных в модель



Генерация файла XML



Генерация кода на языке C



```

- <model name="Drawing1">
- <stateMashine name="A0" description="Светофор">
  <NodeDescriptions />
- <state name="" description="3.Красный">
  <connection source="3.Красный" target="1.Желтый" condition="e1" actions="z1" />
  </state>
- <state name="" description="1.Желтый">
  <connection source="1.Желтый" target="2.Зеленый" condition="e2" actions="z2" />
  <connection source="1.Желтый" target="0.Выключен" condition="x1 > 12.00"
  actions="z4" />
  </state>
- <state name="" description="2.Зеленый">
  <connection source="2.Зеленый" target="3.Красный" condition="e3" actions="z3" />
  <connection source="2.Зеленый" target="0.Выключен" condition="x1 > 7.00"
  actions="z5" />
  </state>
  </stateMashine>
</model>
    
```

```

18: switch (g_AutomataData.y_A0)
19: {
20:
21:     case 0:
22:         /*3.Красный*/
23:         Automaton_A0 (Event);
24:         if (Event == 0)
25:         {
26:
27:             g_AutomataData.y_A0 = 1;
28:         }
29:         break;
30:
31:     case 1:
32:         /*1.Желтый*/
33:         Automaton_A0 (Event);
34:         if (Event == 1)
35:         {
36:
37:
38:             g_AutomataData.y_A0 = 2;
    
```

ГЕНЕРАЦИЯ МОДЕЛИ ДАННЫХ

- ▶ Процесс преобразования – однократный

Создание объекта автомата



Перебор всех фигур на странице

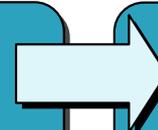
Определение типа фигуры

Если состояние

Добавление в список состояний

Если переход

Добавление в список переходов

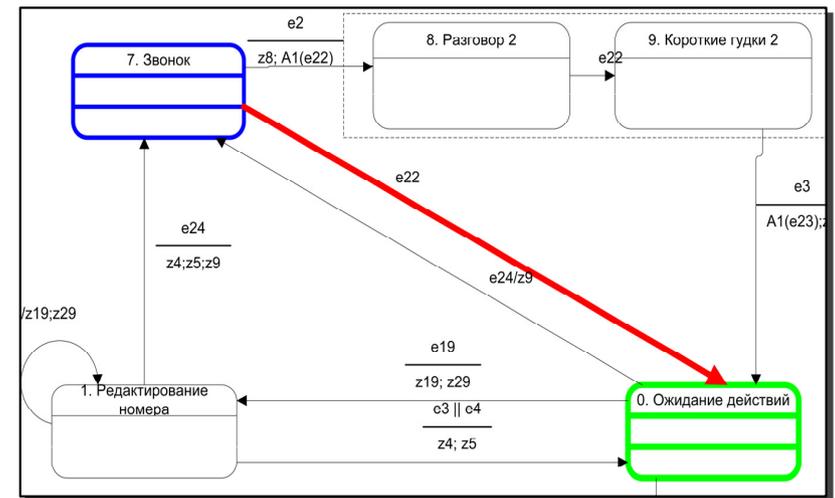


Перебор всех переходов

Хеш-поиск состояния, из которого происходит переход

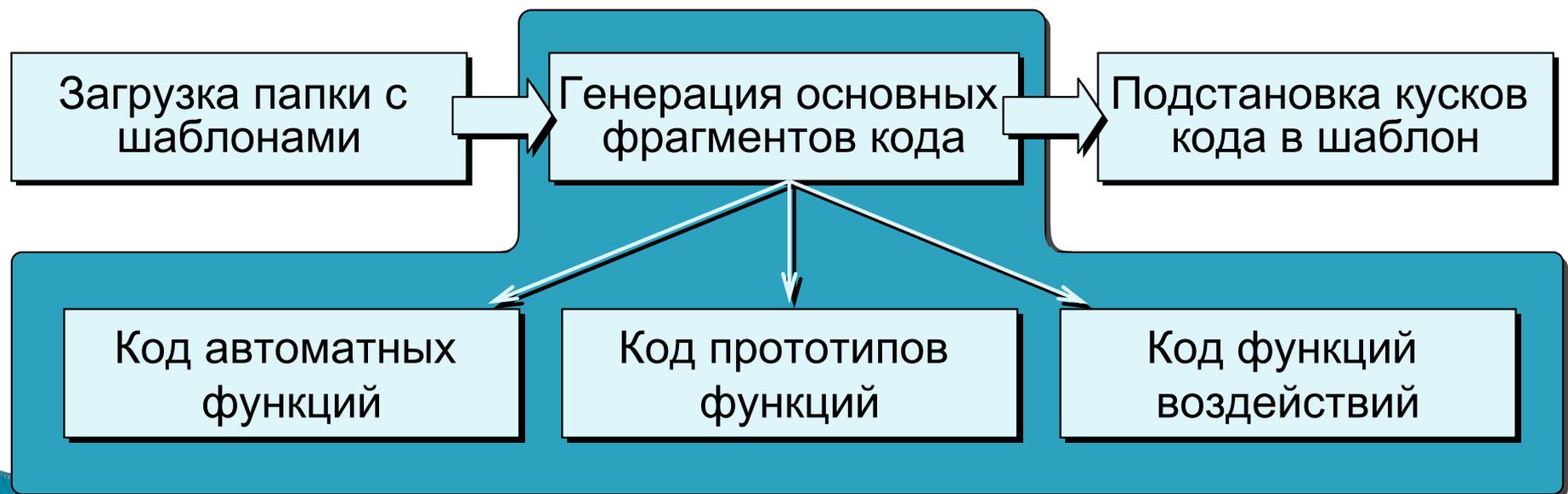


Добавление перехода в список переходов исходного состояния



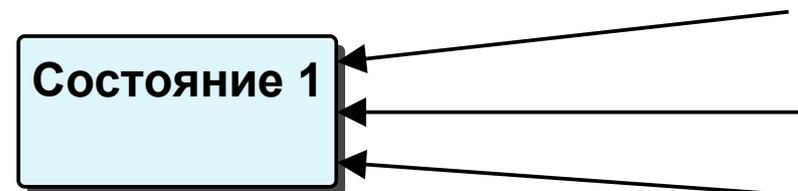
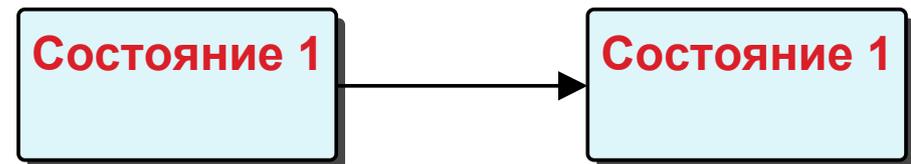
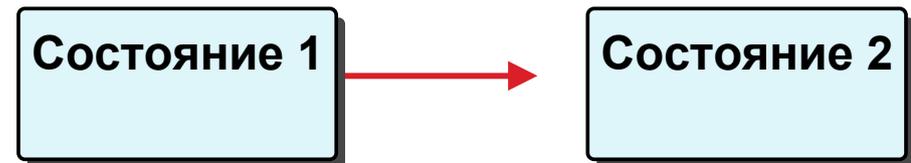
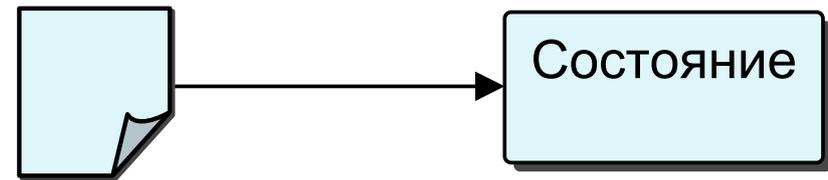
ГЕНЕРАЦИЯ ФАЙЛОВ С КОДОМ

- ▶ Генератор устроен так, что генерацию основных частей исходного кода производит не самостоятельно, а обращаясь к объекту генерации кода
- ▶ Генерация происходит с использованием набора файлов-шаблонов



ТИПЫ ОПРЕДЕЛЯЕМЫХ ОШИБОК

- ▶ Переход соединяет не состояния, а другие фигуры
- ▶ Переход не присоединён к фигуре
- ▶ В автомате найдены одинаковые элементы
- ▶ Из состояния не исходит ни один переход



ПРИМЕР РАБОТЫ ТРАНСЛЯТОРА

The image illustrates the work of a translator by showing the conversion of a C++ program into a Visio flowchart. The C++ code on the left defines a state machine with states 0 (Выключен), 1 (Желтый), 2 (Зеленый), and 3 (Красный). The Visio diagram on the right maps these states to flowchart nodes and transitions. The output window at the bottom shows the build process.

```
break;

case 2:
/*2.Зеленый*/
Automaton_A0 (Ev
if (Event == 2)
{
g_AutomataDa
}
break;

case 3:
/*0.Выключен*/
Automaton_A0 (Ev
if (x1() > 7.00)
{
g_AutomataDa
}
break;
```

The Visio flowchart shows four states: 0. Выключен, 1. Желтый, 2. Зеленый, and 3. Красный. Transitions are labeled e1, e2, z1, z2, z3, and z4. The diagram also includes a table for the automaton's parameters:

Автомат	
Имя:	A0

The output window shows the build process:

```
Show output from: Build
1>----- Build started: Project: AutomatonTest, Configur
1>===== Translation started. =====
1>Starting Visio ...
1>Opening project...
1>Compiling ...
1>Writing XML file ...
1>Deleting old generated files...
```

РЕЗУЛЬТАТЫ

- ▶ Разработана версия программы для трансляции описаний автоматов, представленных в формате Visio, в код на C
 - ▶ Разработано сохранение исходных данных в файл XML
 - ▶ Разработаны средства диагностики ошибок в исходных данных
 - ▶ Разработаны средства интеграции в автоматические процессы сборки приложений (make-файлы, MS Visual Studio build system)
- 